

MILESTONE 2: UPDATE PROJECT PROGRESS

# SMART WATER MANAGEMENT ASSISTANT

---

WITH CLOUD COMPUTING



# Agenda



- 1 Overview
- 2 5 Vs Analysis
- 3 Architecture Plan & Data Analysis  
Technologies, Services and Tools
- 4 Implementation & Results

# Overview

- Thailand is facing flooding every year, especially during the rainy season.
- If there is no effective management, it will cause damage to people's lives, properties, and houses.
- For example, the flooding in 2024 in many areas and the reservoirs could not hold the water until they burst, such as Maha Sarakham and Lampang provinces, etc.

## Objectives

- To develop a model to classify the risk of flooding in areas from rainfall data, water levels in reservoirs and dams.
- The system can display results in real time and forward decision-making recommendations to relevant agencies, such as the Royal Irrigation Department or dam officials.
- We hope this approach will help water management be more precise, reduce impacts on people, and serve as a model for using information technology to help manage water resources.



รูปอ้างเก็บข้าแตกที่ จังหวัดมหาสารคาม พ.ศ. 2567

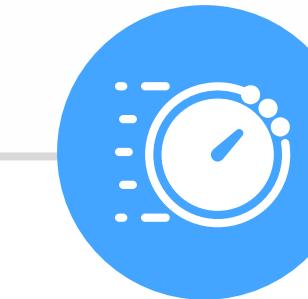
# 5Vs Analysis (1/2)



**Volum**

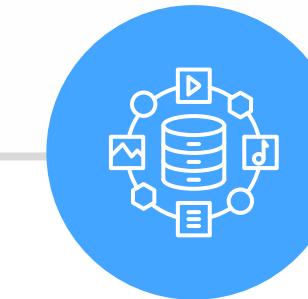
There are a lot of data sources, including.

- Water level of reservoirs from the Royal Irrigation Department.
- Water level of dams from the Royal Irrigation Department.
- Risk area dataset from the Water Resources Information Institute.



**Velocity**

Streaming data from IoT sensors that measure water, such as rainfall and water levels in reservoirs and dams, is received in near real-time, requiring the system to be able to support rapid and continuous data flow.



**Variety**

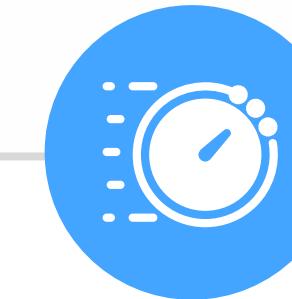
The imported data comes in a variety of formats, such as a structured CSV file of the flood risk dataset and a single JSON file from the Sensor API. The semi-structured data requires data preprocessing and data transformation to format the data into a structure ready for analysis.

# 5Vs Analysis (2/2)



**Veracity**

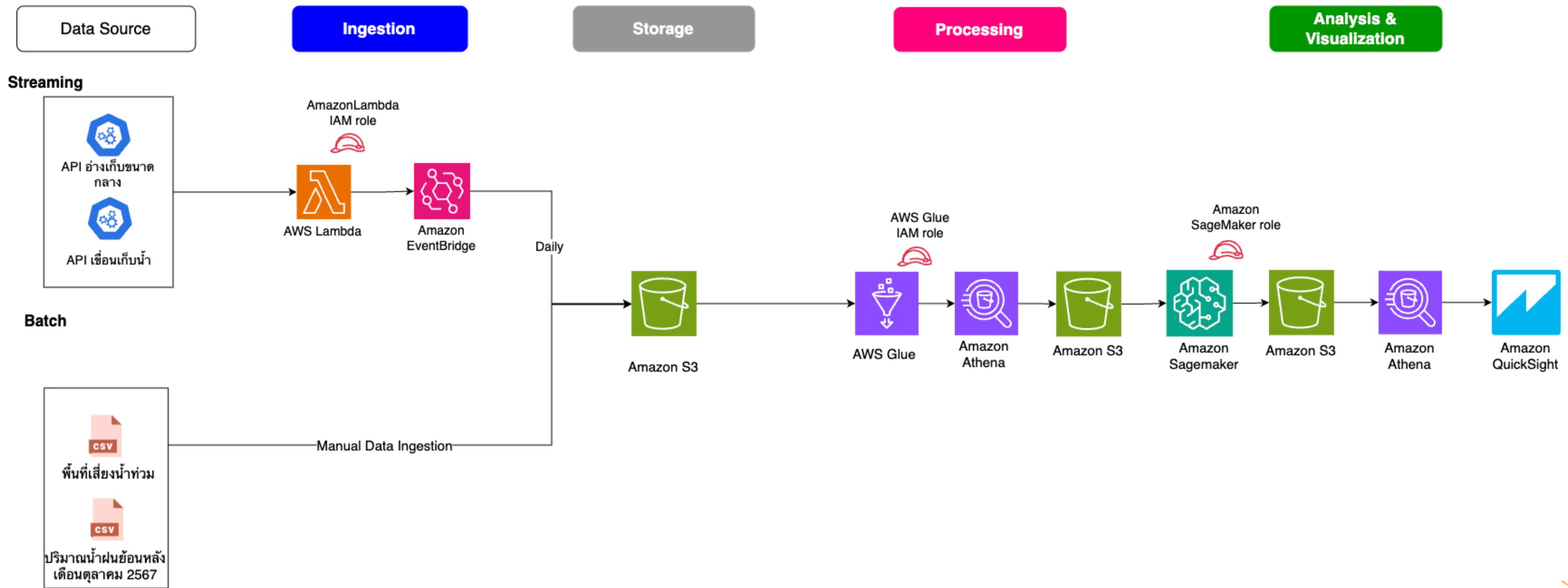
There are steps of Data Cleaning, Outlier Detection and appropriate feature selection to reduce model errors, such as selecting only sensors with reliable data or using imputation techniques for missing data from combining multiple data sources to create training data for the model.



**Value**

This data, when analyzed and modeled, can help predict flood risk areas, which is very valuable for management, such as water management in dams to reduce impacts, evacuation planning, and allowing relevant agencies to make accurate and timely decisions.

# Architecture Plan & Data Analysis



# Technologies, Services and Tools

## Data sources

### Ingestion

#### Streaming Data Ingestion



##### AWS Lambda

To trigger streaming data that API Gateway received. And then, automate transfers data to store in Amazon S3.



##### Amazon EventBridge

To trigger Lambda function to get API information and send it to S3 bucket daily.



##### IAM role

To allow Lambda function put information to keep in S3 Bucket.

#### Batch Data Ingestion



##### Manual

Because the dataset of the data source is the history of dataset from the past.

### Storage



#### Amazon S3

To store raw data from API Gateway (streaming data) and raw dataset (batch data) in S3 buckets.

Moreover, we use S3 bucket to store the data after the processing.

### Processing

#### Streaming & Batch processing



##### AWS Glue

To create Glue crawler to extract, transform, and load data into an AWS Glue database. Editing schema of the table.

Moreover, creating Glue job and a trigger to run the job daily.



##### IAM role

To allow AWS Glue access S3 Bucket and Amazon Athena.



##### Amazon Athena

To preview and query data to clean, select and create the table that we need to use in the next step. .



### Analysis & Visualization

#### Analysis



##### Amazon SageMaker

To use Amazon SageMaker analysis and predict the possibility of flooding areas with near-real-time.



##### IAM role

To allow Amazon SageMaker access S3 Bucket.



##### Amazon Athena

To query data and Amazon Athena plugin .

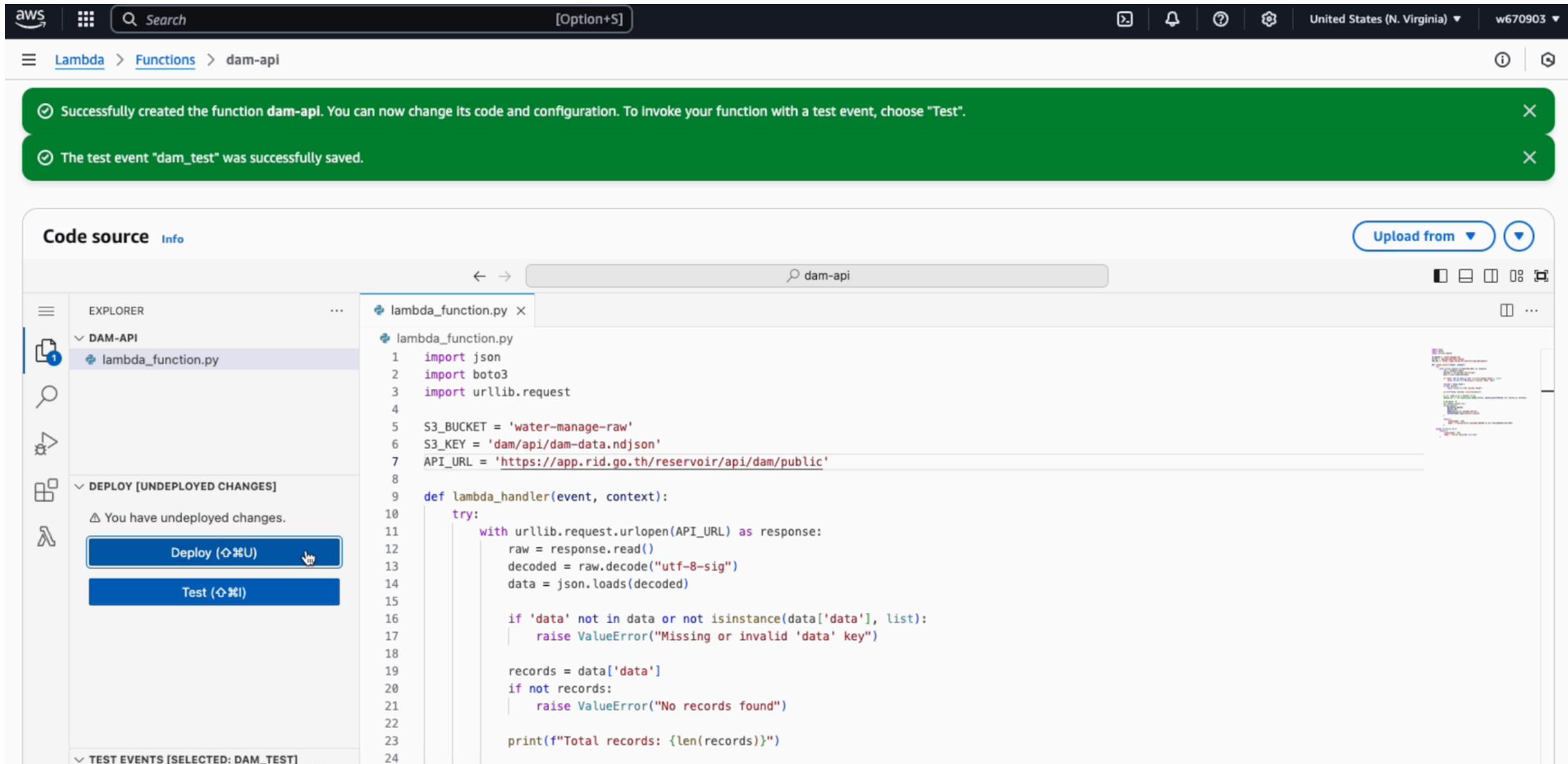


#### Visualization

##### Amazon QuickSight

To creates a dashboard to displays the information after we analysed the cleaned data by Amazon SageMaker. We hope these data provide the advantages for related people and others who are interested in it to manage water in dam and prepare plan to handle with it before the flooding.

# Implementation : Ingestion (1/3)



The screenshot shows the AWS Lambda function editor for a function named "dam-api". The code source is "lambda\_function.py". The code imports json, boto3, and urllib.request, and defines a lambda\_handler function that reads data from an S3 bucket, parses it, and prints the total number of records. A test event named "dam\_test" has been successfully saved.

```
import json
import boto3
import urllib.request

S3_BUCKET = 'water-manage-raw'
S3_KEY = 'dam/api/dam-data.ndjson'
API_URL = 'https://app.rid.go.th/reservoir/api/dam/public'

def lambda_handler(event, context):
    try:
        with urllib.request.urlopen(API_URL) as response:
            raw = response.read()
            decoded = raw.decode("utf-8-sig")
            data = json.loads(decoded)

            if 'data' not in data or not isinstance(data['data'], list):
                raise ValueError("Missing or invalid 'data' key")

            records = data['data']
            if not records:
                raise ValueError("No records found")

            print(f"Total records: {len(records)}")
```

## Create Lambda Function to import the water level's API

# Implementation : Ingestion (2/3)

The screenshot shows two separate browser tabs for the AWS Lambda console.

**Left Tab:** The user is creating a new schedule named "lambda-dam-daily-api". The configuration includes:

- Schedule detail:** Schedule name: lambda-dam-daily-api, Description: -, Schedule group: default.
- Occurrence:** Recurring.
- Start date and time:** -
- Flexible time window:** Off.
- Cron expression:** 0 \* \* \* ? \* (Minutes, Hours, Day of month, Month, Day of week, Year).
- Next 10 trigger dates:** Displays the next 10 scheduled times in UTC+07:00 (Asia/Bangkok), starting from Thu, 29 May 2025 01:00:00.

**Right Tab:** The user is creating a new schedule named "lambda-reservoir-daily-api". The configuration includes:

- Schedule detail:** Schedule name: lambda-reservoir-daily-api, Description: -, Schedule group: default.
- Occurrence:** Recurring.
- Start date and time:** -
- Flexible time window:** Off.
- Cron expression:** 0 \* \* \* ? \* (Minutes, Hours, Day of month, Month, Day of week, Year).
- Next 10 trigger dates:** Displays the next 10 scheduled times in UTC+07:00 (Asia/Bangkok), starting from Thu, 05 Jun 2025 01:00:00.

**Common UI Elements:** Both tabs show the AWS Lambda console header with tabs for IAM, Functions, Lambda, and Reservoir API. The sidebar on the right contains sections for Developer resources, Buses, Pipes, Scheduler, Integration, and Schema registry. A summary message at the top of each tab indicates the schedule is being created.

## Create EventBridge Schedule to trigger Lambda Function

# Implementation : Ingestion (3/3)

The image consists of two side-by-side screenshots of the AWS S3 console.

**Left Screenshot (Upload Objects):** This screenshot shows the process of uploading a file to an S3 bucket. The URL is `us-east-1.console.aws.amazon.com/s3/upload/water-manage-raw?region=us-east-1&bucketType=general&prefix=risk-area/raw/`. The 'Objects' tab is selected. A table lists one file: `flood-risk-area.csv`, which is a `text/csv` file, 5.8 MB in size. Below the table, sections for 'Destination' (set to `s3://water-manage-raw/risk-area/raw/`) and 'Permissions' are visible. At the bottom right is an orange 'Upload' button.

**Right Screenshot (Bucket Objects):** This screenshot shows the contents of the `risk-area/raw/` folder in the `water-manage-raw` bucket. The URL is `us-east-1.console.aws.amazon.com/s3/buckets/water-manage-raw?region=us-east-1&bucketType=general&prefix=risk-area%2Fraw%2F&tab=objects`. The 'Objects' tab is selected, showing the same `flood-risk-area.csv` file listed. The file's properties are shown: it was last modified on May 28, 2025, at 10:59:48 (UTC+07:00), is 5.8 MB in size, and has a storage class of Standard. An orange 'Copy S3 URI' button is located at the top right of the object list.

Manual upload “risk area.csv” file to S3 bucket.

# Implementation : Storage

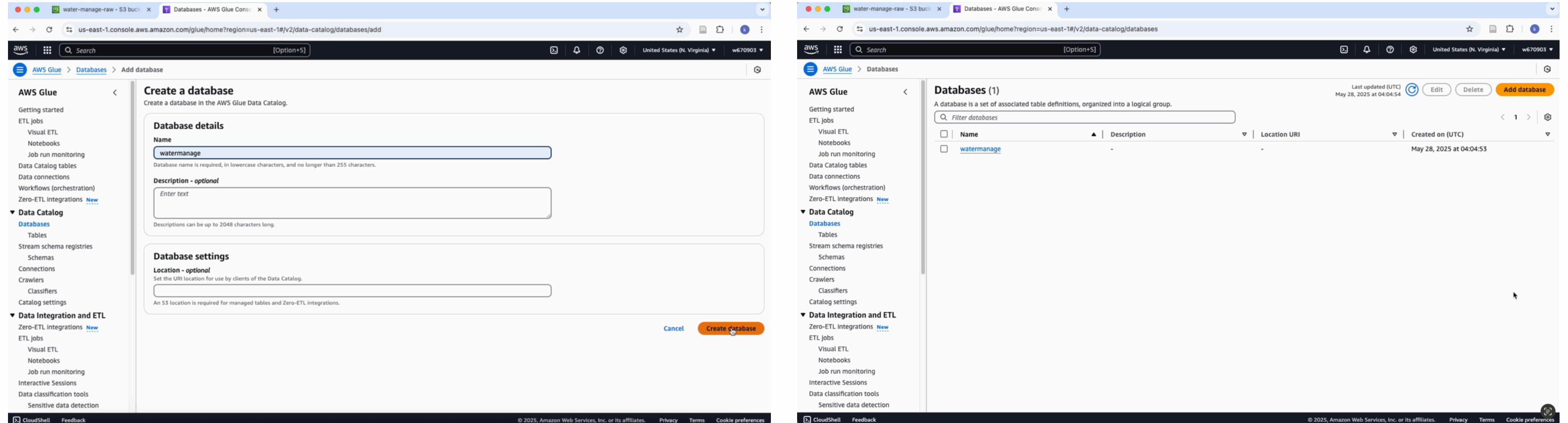
The image contains two side-by-side screenshots of the AWS S3 console.

**Left Screenshot:** Shows the 'Buckets' page. A green success message at the top states 'Successfully created bucket "water-manage-processed"'. Below it is an 'Account snapshot' section with an 'Update every 24 hours' button. Under 'General purpose buckets', there are two buckets listed: 'water-manage-processed' and 'water-manage-raw'. Both buckets were created on May 28, 2025, at 10:43:55 UTC+07:00. The 'Create bucket' button is visible at the top right of the list.

**Right Screenshot:** Shows the contents of the 'water-manage-raw' bucket. A green success message at the top states 'Successfully created folder "rainfall!"'. The 'Objects' tab is selected, showing four objects/folders: 'dam/' (Folder), 'rainfall/' (Folder), 'reservoir/' (Folder), and 'risk-area/' (Folder). The 'Actions' menu is open, showing options like Copy S3 URI, Copy URL, Download, Open, Delete, Actions, Create folder, and Upload.

Create S3 Buckets and folders to keep raw data and processed data

# Implementation : Processing (1/5)



The image consists of two side-by-side screenshots of the AWS Glue Data Catalog interface.

**Left Screenshot (Create a database):**

- Header:** water-manage-raw - S3 buck, Databases - AWS Glue Cons...
- Breadcrumbs:** AWS Glue > Databases > Add database
- Title:** Create a database
- Form Fields:**
  - Name:** watermanage
  - Description - optional:** Enter text
  - Database settings:**
    - Location - optional:** Set the URI location for use by clients of the Data Catalog.
- Buttons:** Cancel, Create database

**Right Screenshot (Databases list):**

- Header:** water-manage-raw - S3 buck, Databases - AWS Glue Cons...
- Breadcrumbs:** AWS Glue > Databases
- Title:** Databases (1)
- Table:** Displays the created database.

| Name        | Description | Location URI | Created on (UTC)         |
|-------------|-------------|--------------|--------------------------|
| watermanage | -           | -            | May 28, 2025 at 04:04:53 |
- Buttons:** Edit, Delete, Add database

Create Glue database for supporting and managing metadata and schema.

# Implementation : Processing (2/5)

The image contains two side-by-side screenshots of the AWS Glue console.

**Left Screenshot (Crawlers - AWS Glue Console):** This screenshot shows the "Add crawler" wizard. The steps are:

- Step 1: Set crawler properties**: Name is set to "risk-area-raw".
- Step 2: Choose data sources and classifiers**: Data source is set to "S3" pointing to "s3://water-manage-raw/risk-area/raw/".
- Step 3: Configure security settings**: IAM role is "AWSGlueServiceRole-water".
- Step 4: Set output and scheduling**: Database is "watermanage".

**Right Screenshot (Tables - AWS Glue Console):** This screenshot shows the "Edit schema" dialog for a table named "raw". It displays the current schema entries:

| #  | Name        | Type   | Comment (optional) |
|----|-------------|--------|--------------------|
| 1  |             | -      | -                  |
| 2  |             | -      | -                  |
| 3  |             | -      | -                  |
| 4  |             | -      | -                  |
| 5  |             | -      | -                  |
| 6  |             | -      | -                  |
| 7  |             | -      | -                  |
| 8  |             | -      | -                  |
| 9  | province_th | string | -                  |
| 10 |             | -      | -                  |
| 11 |             | -      | -                  |
| 12 |             | -      | -                  |
| 13 | risk        | string | -                  |

A modal dialog is open for "Edit schema entry" for column #9, with the name "province\_th", type "string", and a comment field.

Create table by using crawler and edit schema of risk area

# Implementation : Processing (3/5)

The screenshot shows the 'Add crawler' wizard in the AWS Glue console. The steps are:

- Step 1: Set crawler properties**: Name is 'risk-area-raw'. Description and Tags are empty.
- Step 2: Choose data sources and classifiers**: Data source is 'S3' pointing to 's3://water-manage-raw/risk-area/raw/'. Parameters include 'Recrawl all'.
- Step 3: Configure security settings**: IAM role is 'AWSGlueServiceRole-water'. Security configuration and Lake Formation configuration are empty.
- Step 4: Set output and scheduling**: Database is 'watermanage'. Table prefix is empty. Maximum table threshold is empty. Schedule is 'On demand'.

At the bottom right are 'Cancel', 'Previous', and 'Create crawler' buttons.

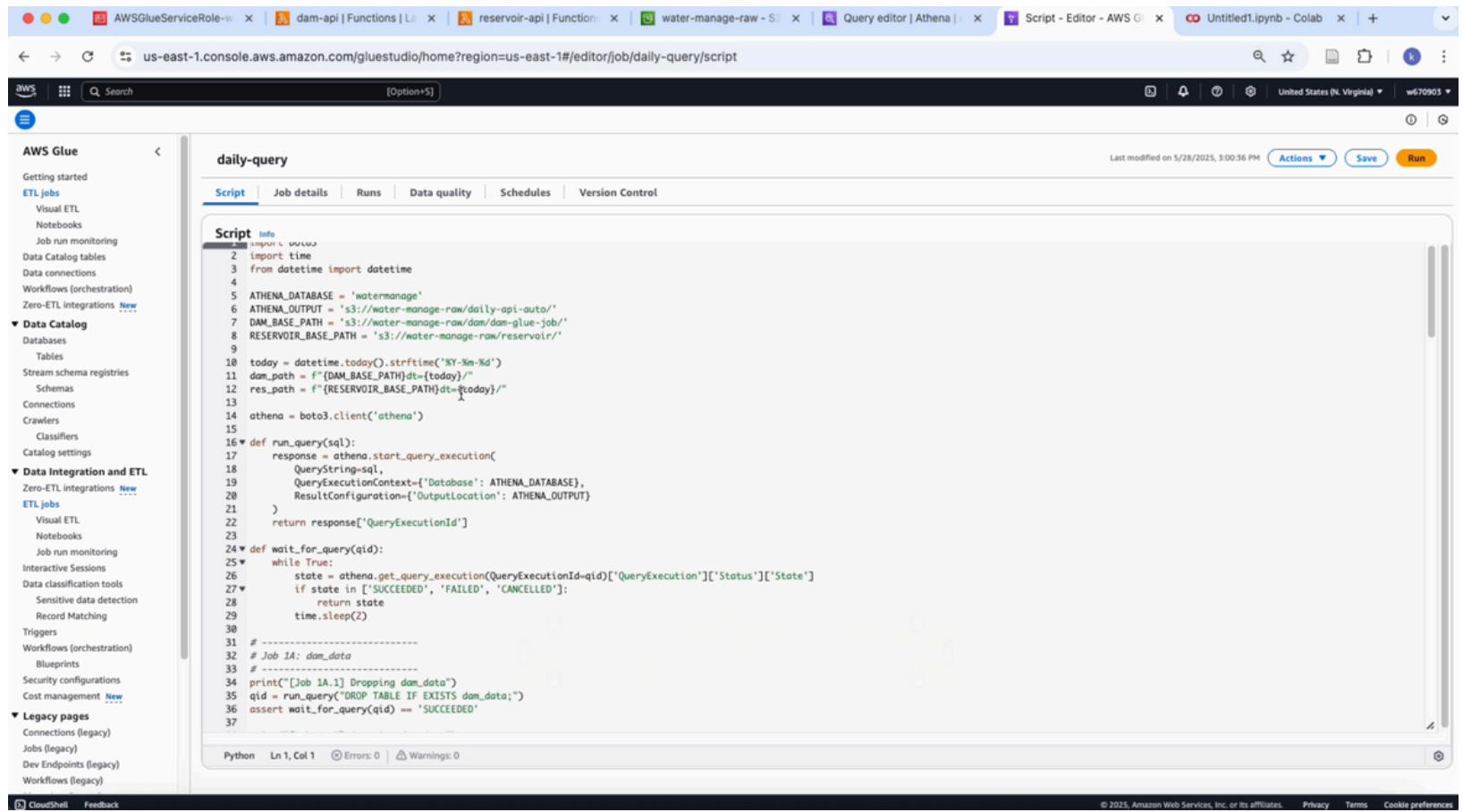
The screenshot shows the 'Edit schema' dialog for the 'raw' table. It displays the current schema:

| #  | Name        | Type   | Comment (optional) |
|----|-------------|--------|--------------------|
| 1  | -           | -      | -                  |
| 2  | -           | -      | -                  |
| 3  | -           | -      | -                  |
| 4  | -           | -      | -                  |
| 5  | -           | -      | -                  |
| 6  | -           | -      | -                  |
| 7  | -           | -      | -                  |
| 8  | -           | -      | -                  |
| 9  | province_th | string | -                  |
| 10 | -           | -      | -                  |
| 11 | -           | -      | -                  |
| 12 | -           | -      | -                  |
| 13 | risk        | string | -                  |

A modal dialog is open for 'Edit schema entry' for column #9, with fields: Name 'province\_th', Data type 'string', and Comment 'criteria'.

Create table by using crawler and edit schema of risk area

# Implementation : Processing (4/5)



The screenshot shows the AWS Glue Studio interface. On the left, there's a sidebar with various AWS services like AWS Glue, Data Catalog, and Data Integration and ETL. The main area displays a Python script titled 'daily-query'. The script imports time, datetime, and boto3, sets environment variables for ATHENA\_DATABASE, DAM\_BASE\_PATH, and RESERVOIR\_BASE\_PATH, and defines functions for running queries and waiting for their completion. It also includes logic for dropping a table named 'dam\_data' if it exists.

```
ATHENA_DATABASE = 'water-manage'
DAM_BASE_PATH = 's3://water-manage-raw/dam/dam-glue-job/'
RESERVOIR_BASE_PATH = 's3://water-manage-raw/reservoir/'

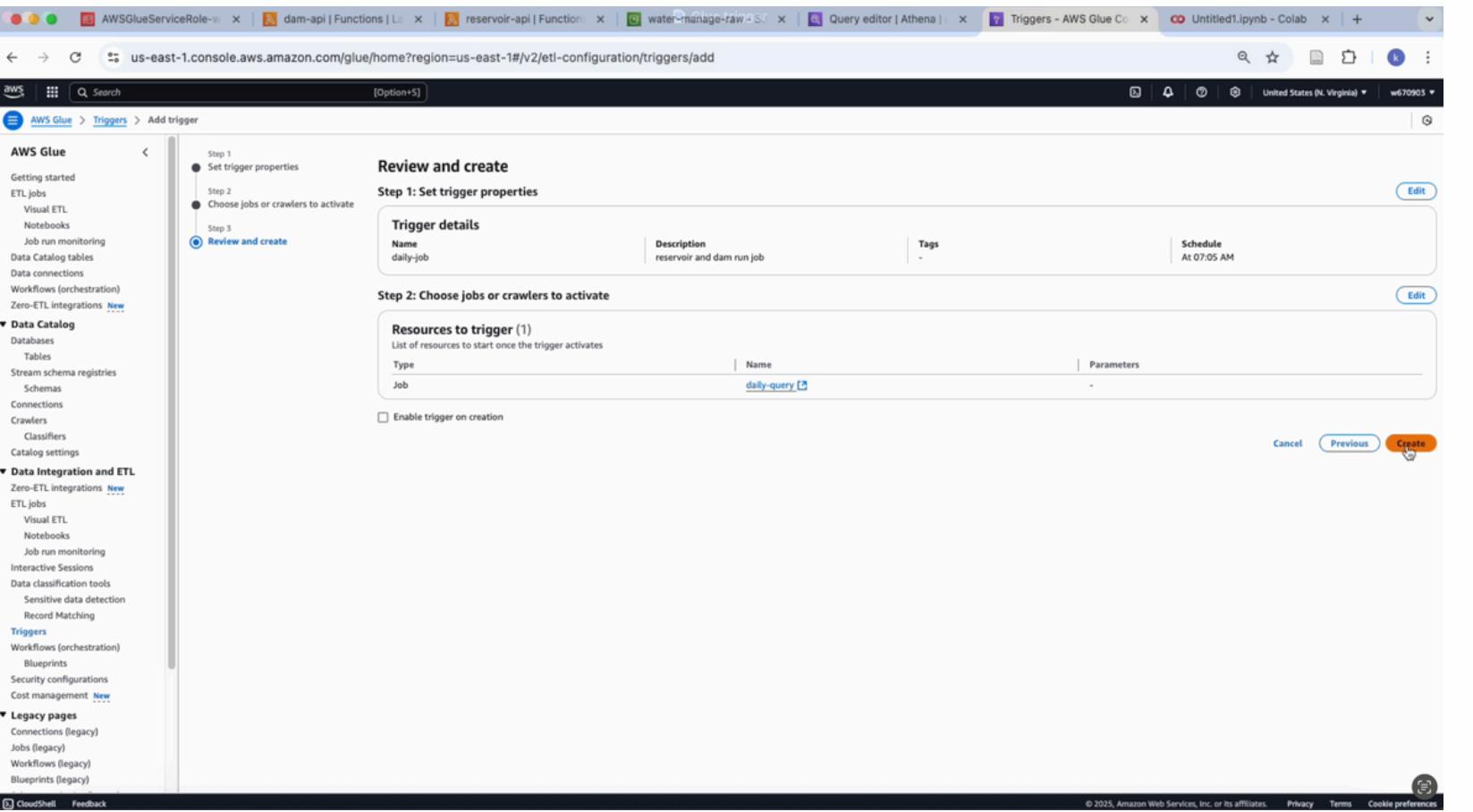
today = datetime.date.today().strftime("%Y-%m-%d")
dam_path = f'{DAM_BASE_PATH}{today}/'
res_path = f'{RESERVOIR_BASE_PATH}{today}/'

athena = boto3.client('athena')

def run_query(sql):
    response = athena.start_query_execution(
        QueryString=sql,
        QueryExecutionContext={'Database': ATHENA_DATABASE},
        ResultConfiguration={'OutputLocation': ATHENA_OUTPUT}
    )
    return response['QueryExecutionId']

def wait_for_query(qid):
    while True:
        state = athena.get_query_execution(QueryExecutionId=qid)['QueryExecution']['Status']['State']
        if state in ['SUCCEEDED', 'FAILED', 'CANCELLED']:
            return state
        time.sleep(2)

# -----
# Job 1A: dam_data
# -----
print("[Job 1A] Dropping dam_data")
qid = run_query("DROP TABLE IF EXISTS dam_data;")
assert wait_for_query(qid) == "SUCCEEDED"
```



Create Glue job by using script for streaming data and add trigger for running Glue job

# Implementation : Processing (5/5)

The image displays two side-by-side screenshots of the Amazon Athena Query Editor interface.

**Left Screenshot:** Shows the Query editor interface with the following details:

- Data Source:** AwsDataCatalog
- Catalog:** None
- Database:** watermanage
- Tables and views:** raw
- Query 1:** A CREATE TABLE statement for "riskarea\_processed". The statement includes creating a table with columns month, province, amphoe, tambon, risk\_label, and location\_key. It uses TRIM(REPLACE) functions to clean up province, amphoe, and tambon names. It also includes a CASE statement for risk\_label.
- Query results:** No results displayed.

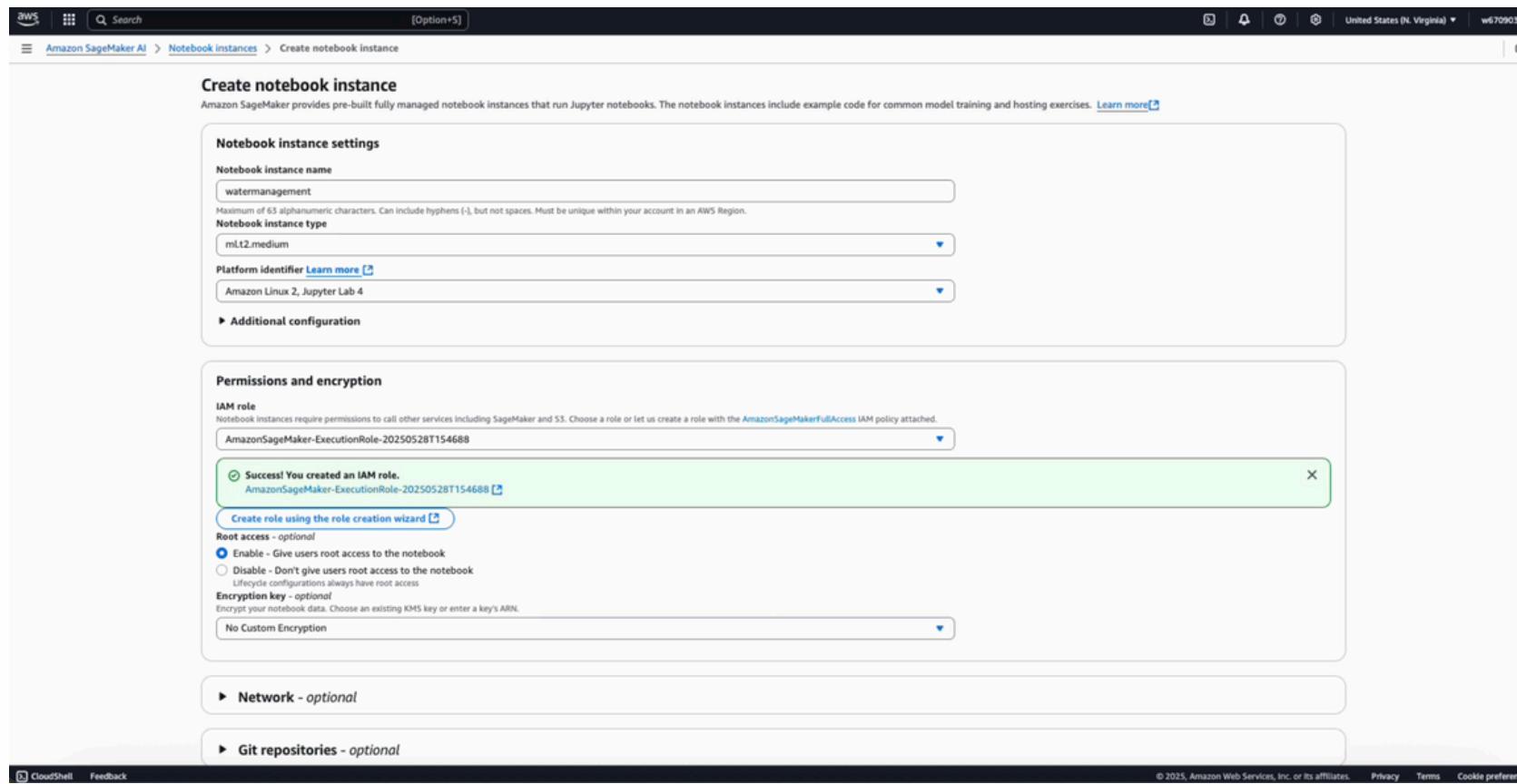
**Right Screenshot:** Shows the results of the query execution:

- Run again:** Run again
- Completed:** Time in queue: 108 ms, Run time: 449 ms, Data scanned: 196.50 KB
- Results (10):** Displays 10 rows of data from the "riskarea\_processed" table.

| #  | month | province | amphoe    | tambon    | risk_label | location_key            |
|----|-------|----------|-----------|-----------|------------|-------------------------|
| 1  | 5     | แพรฯ     | เมืองแพรฯ | แม่ค่ามี  | 0          | แพรฯ_เมืองแพรฯ_แม่ค่ามี |
| 2  | 5     | แพรฯ     | เมืองแพรฯ | ทุ่งดาว   | 0          | แพรฯ_เมืองแพรฯ_ทุ่งดาว  |
| 3  | 5     | แพรฯ     | เมืองแพรฯ | ท่าข้าม   | 0          | แพรฯ_เมืองแพรฯ_ท่าข้าม  |
| 4  | 5     | แพรฯ     | เมืองแพรฯ | แม่ยอม    | 0          | แพรฯ_เมืองแพรฯ_แม่ยอม   |
| 5  | 5     | แพรฯ     | รังกวาล   | แม่ยาจดา  | 0          | แพรฯ_รังกวาล_แม่ยาจดา   |
| 6  | 5     | แพรฯ     | รังกวาล   | แม่ยาจ่อง | 0          | แพรฯ_รังกวาล_แม่ยาจ่อง  |
| 7  | 5     | แพรฯ     | สูบเม่น   | สูบเม่น   | 0          | แพรฯ_สูบเม่น_สูบเม่น    |
| 8  | 5     | แพรฯ     | สูบเม่น   | น้ำด่า    | 0          | แพรฯ_สูบเม่น_น้ำด่า     |
| 9  | 5     | แพรฯ     | สูบเม่น   | หัว่าย    | 0          | แพรฯ_สูบเม่น_หัว่าย     |
| 10 | 5     | แพรฯ     | สูบเม่น   | ตอนมูล    | 0          | แพรฯ_สูบเม่น_ตอนมูล     |

To use Amazon Athena to query and clean data

# Implementation : Analysis and Visualization (1/2)



The screenshot shows a Jupyter Notebook titled 'K-cluster.ipynb'. The code is organized into cells numbered 7 through 15. Cells 7 and 8 contain imports and initial setup for S3 and data loading. Cells 9 and 10 show the creation of an S3 client. Cells 11 and 12 load data from an S3 bucket. Cell 13 handles missing data. Cells 14 and 15 perform clustering and output results.

```
[7]: # STEP 1: Import Libraries
import boto3
import pandas as pd
import numpy as np
from io import StringIO
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.metrics import classification_report, confusion_matrix, ConfusionMatrixDisplay
from scipy.stats import mode
import matplotlib.pyplot as plt
from IPython.display import display

[8]: # STEP 2: S3 Configuration
input_bucket = 'water-manage-raw'
input_key = 'October-2024/risk_with_all_pct.csv'
output_bucket = 'water-manage-processed'
output_key = 'k-cluster/kmeans_results.csv'
region = 'us-east-1'

[9]: s3 = boto3.client('s3', region_name=region)

[10]: # STEP 3: Load data from S3
response = s3.get_object(Bucket=input_bucket, Key=input_key)
df = pd.read_csv(response['Body'])

# STEP 4 (Optional): Fill Missing and Handle Scaling
features = [
    'res_avg_volume', 'res_avg_percent_storage', 'res_avg_inflow', 'res_avg_outflow',
    'res_dam_avg_volume', 'res_dam_avg_percent_storage', 'res_dam_inflow', 'res_dam_avg_outflow',
    'res_avg_rainfall_mm'
]

# สำหรับ field ที่จะเก็บข้อมูล -> ถ้าไม่ -> ให้ลบ
res_fields = ['res_avg_volume', 'res_avg_percent_storage', 'res_avg_inflow', 'res_avg_outflow']
dam_fields = ['res_dam_avg_volume', 'res_dam_avg_percent_storage', 'res_dam_inflow', 'res_dam_avg_outflow']
```

## Create SageMaker Notebook and analysis data.

# Implementation : Analysis and Visualization (2/2)



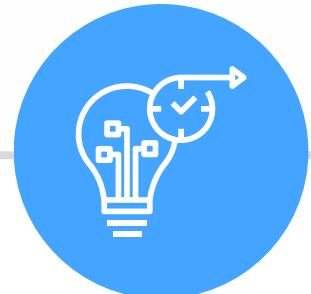
# Visualize

# Result and future work



## Result

- End-to-end data pipeline developed on AWS (Lambda, Glue, Athena, SageMaker)
- Multi-source data integration (API JSON + CSV from public agencies)
- ML model predicts flood risk at sub-district level with high accuracy
- Real-time dashboard built with Grafana for spatial visualization
- Identified key challenges in data standardization and governance
- Gained practical insights on translating technical results for decision-makers



## Future Work

- Integrate weather forecast data (GFS, ECMWF) for real-time prediction
- Build Early Warning System with mobile notifications (LINE/SMS)
- Apply online learning for adaptive model updates
- Improve data quality via imputation and outlier detection
- Develop interactive flood risk maps (drill-down to tambon level)
- Design user-friendly tools to support local and community-level use





# APPENDIX

Milestone 2 update project progress

# Data cleaning or transformation steps & How the pipeline handles data quality or noise



**Batch**

## **Glue Clawler**

- Edit column name for schema.

## **Amazon Athena**

- Query data in the table (CSV file) to preview the information, i.e., null, blank, N/A.
- Query to create the table that delete the data in the table that contain null, blank, N/A.
- Query to select columns that we would like to use in the analysis.
- Transform risk column from “string” to “integer”.



**Streaming**

## **Amazon Athena**

- Query data in (CSV file) that transform from JSON (API) to preview the information, i.e., null, blank, N/A.
- Query to create the table that delete the data in the table that contain null, blank, N/A or outlier data.
- Eliminate duplicate data.
- Query to select columns that we would like to use in the analysis.

AWS | Search [Option+S] | United States (N. Virginia) | vclabs/user3452026=6709036013\_Kawisara\_Wsootsakunsak @ 8359... ▾

AWS Glue > Crawlers > riskarea

Crawler successfully starting  
The following crawler is now starting: "riskarea"

riskarea

Last updated (UTC)  
May 2, 2025 at 09:20:34

Run crawler Edit Delete

**Crawler properties**

|  |                             |                                   |                   |
|--|-----------------------------|-----------------------------------|-------------------|
| Name<br>riskarea                             | IAM role<br>LabRole         | Database<br>watermanage           | State<br>READY    |
| Description<br>The risk flooding area record | Security configuration<br>- | Lake Formation configuration<br>- | Table prefix<br>- |
| Maximum table threshold<br>-                 |                             |                                   |                   |

► Advanced settings

**Crawler runs** | Schedule | Data sources | Classifiers | Tags

**Crawler runs (1)**  
The list of crawler runs for this crawler.

| Start time (UTC)        | End time (UTC)          | Current/last duration | Status    | DPU hours | Table changes |
|-------------------------|-------------------------|-----------------------|-----------|-----------|---------------|
| May 2, 2025 at 09:20:44 | May 2, 2025 at 09:22:04 | 01 min 19 s           | Completed | -         | -             |

Stop run View CloudWatch logs View run details

Filter data Filter by a date and time range

< 1 > | ⚙

Interactive Sessions

# Create Glue Clawler

# Before

Schema | Partitions | Indexes | Column statistics - new

**Schema (13)**  
View and manage the table schema.

| #  | Column name   | Data type | Partition key | Comment |
|----|---------------|-----------|---------------|---------|
| 1  | month         | bigint    | -             | -       |
| 2  | geocode       | bigint    | -             | -       |
| 3  | tambon_t      | string    | -             | -       |
| 4  | tambon_e      | string    | -             | -       |
| 5  | amphoe_code   | bigint    | -             | -       |
| 6  | amphoe_t      | string    | -             | -       |
| 7  | amphoe_e      | string    | -             | -       |
| 8  | prov_code     | bigint    | -             | -       |
| 9  | prov_t        | string    | -             | -       |
| 10 | prov_e        | string    | -             | -       |
| 11 | count 17 year | bigint    | -             | -       |
| 12 | criteria      | string    | -             | -       |
| 13 | risk          | string    | -             | -       |

Edit schema as JSON | Edit schema



# After

Schema | Partitions | Indexes | Column statistics - new

**Schema (13)**  
View and manage the table schema.

| #  | Column name     | Data type | Partition key | Comment |
|----|-----------------|-----------|---------------|---------|
| 1  | month           | bigint    | -             | -       |
| 2  | geocode         | bigint    | -             | -       |
| 3  | sub_district_th | string    | -             | -       |
| 4  | sub_district_en | string    | -             | -       |
| 5  | district_code   | bigint    | -             | -       |
| 6  | district_th     | string    | -             | -       |
| 7  | district_en     | string    | -             | -       |
| 8  | province_code   | bigint    | -             | -       |
| 9  | province_th     | string    | -             | -       |
| 10 | province_eh     | string    | -             | -       |
| 11 | count_17_years  | bigint    | -             | -       |
| 12 | criteria        | string    | -             | -       |
| 13 | risk            | string    | -             | -       |

Edit schema as JSON | Edit schema

# Edit column name and create schema

The screenshot shows the 'Manage settings' page for Amazon Athena's 'Query editor'. The left sidebar includes links for 'Amazon Athena', 'Query editor' (which is selected), 'Notebook editor', 'Notebook explorer', 'Jobs' (with 'Workflows' and 'Powered by Step Functions'), and 'Administration' (with 'Workgroups', 'Capacity reservations', and 'Data sources and catalogs'). A 'What's new' section indicates 9+ updates. A 'Turn on compact mode' toggle is also present. The main content area is titled 'Manage settings' and contains sections for 'Query result location and encryption' and 'Expected bucket owner - optional'. In the 'Query result location and encryption' section, there is a text input field containing 's3://processingdata2025/riskarea\_p', a 'View' button, a 'Browse S3' button, and a 'Lifecycle configuration' button. Below this, there is a note about creating lifecycle rules for the bucket. In the 'Expected bucket owner - optional' section, there is a text input field for 'Enter AWS account ID' and two checkboxes: 'Assign bucket owner full control over query results' and 'Encrypt query results'. At the bottom right are 'Cancel' and 'Save' buttons.

# Select S3 bucket to store query table

[Query results](#) | [Query stats](#)[Completed](#)

Time in queue: 107 ms Run time: 677 ms Data scanned: 816.90 KB

**Results (10)**[Copy](#)[Download results CSV](#) Search rows

&lt; 1 &gt; | ⌂

| #  | month | geocode | sub_district_th | sub_district_en | district_code | district_th       | district_en                       | province_code | province_th       |
|----|-------|---------|-----------------|-----------------|---------------|-------------------|-----------------------------------|---------------|-------------------|
| 1  | 1     | 120501  | ต.ไทรน้อย       | Sai Noi         | 1205          | อ.ไทรน้อย         | Sai Noi District                  | 12            | จ.นนทบุรี         |
| 2  | 1     | 120503  | ต.หนองเพราฯ     | Nong Phrao Ngai | 1205          | อ.ไทรน้อย         | Sai Noi District                  | 12            | จ.นนทบุรี         |
| 3  | 1     | 120504  | ต.ไทรใหญ่       | Sai Yai         | 1205          | อ.ไทรน้อย         | Sai Noi District                  | 12            | จ.นนทบุรี         |
| 4  | 1     | 120505  | ต.ชุมศรี        | Khun Si         | 1205          | อ.ไทรน้อย         | Sai Noi District                  | 12            | จ.นนทบุรี         |
| 5  | 1     | 120507  | ต.ทวีวัฒนา      | Thawi Watthana  | 1205          | อ.ไทรน้อย         | Sai Noi District                  | 12            | จ.นนทบุรี         |
| 6  | 1     | 140110  | ต.สวนพริก       | Suan Phrik      | 1401          | อ.พระนครศรีอยุธยา | Phra Nakhon Si Ayutthaya District | 14            | จ.พระนครศรีอยุธยา |
| 7  | 1     | 140507  | ต.บ้านคลัง      | Ban Khlang      | 1405          | อ.บางนาล          | Bang Ban District                 | 14            | จ.พระนครศรีอยุธยา |
| 8  | 1     | 140509  | ต.น้ำเต้า       | Namtao          | 1405          | อ.บางนาล          | Bang Ban District                 | 14            | จ.พระนครศรีอยุธยา |
| 9  | 1     | 140511  | ต.วัดตาก        | Wat Taku        | 1405          | อ.บางนาล          | Bang Ban District                 | 14            | จ.พระนครศรีอยุธยา |
| 10 | 1     | 140512  | ต.บางหลวง       | Bang Luang      | 1405          | อ.บางนาล          | Bang Ban District                 | 14            | จ.พระนครศรีอยุธยา |

# Preview data in table

Query results

Query stats

Completed

Time in queue: 121 ms

Run time: 514 ms

Data scanned: 115.40 KB

Results (10)

Copy

Download results CSV

Search rows

< 1 >



| #  | month | sub_district_en | district_en                       | province_en            | count_17_years | risk      |
|----|-------|-----------------|-----------------------------------|------------------------|----------------|-----------|
| 1  | 1     | Sai Noi         | Sai Noi District                  | Nonthaburi             | 1              | เสี่ยงต่อ |
| 2  | 1     | Nong Phrao Ngai | Sai Noi District                  | Nonthaburi             | 1              | เสี่ยงต่อ |
| 3  | 1     | Sai Yai         | Sai Noi District                  | Nonthaburi             | 1              | เสี่ยงต่อ |
| 4  | 1     | Khun Si         | Sai Noi District                  | Nonthaburi             | 1              | เสี่ยงต่อ |
| 5  | 1     | Thawi Watthana  | Sai Noi District                  | Nonthaburi             | 1              | เสี่ยงต่อ |
| 6  | 1     | Suan Phrik      | Phra Nakhon Si Ayutthaya District | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ |
| 7  | 1     | Ban Khlang      | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ |
| 8  | 1     | Namtao          | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ |
| 9  | 1     | Wat Taku        | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ |
| 10 | 1     | Bang Luang      | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ |

Only Select table that  
would like to use

⌚ Completed

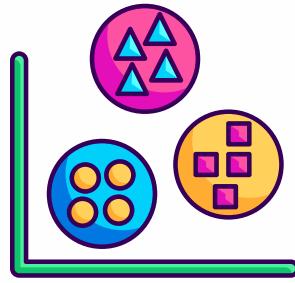
Time in queue: 70 ms Run time: 888 ms Data scanned: 115.40 KB

Results (27,023)

| #  | month | sub_district_en | district_en                       | province_en            | count_17_years | risk      | risk_status | risk_score |
|----|-------|-----------------|-----------------------------------|------------------------|----------------|-----------|-------------|------------|
| 1  | 1     | Sai Noi         | Sai Noi District                  | Nonthaburi             | 1              | เสี่ยงต่อ | low         | 1          |
| 2  | 1     | Nong Phrao Ngai | Sai Noi District                  | Nonthaburi             | 1              | เสี่ยงต่อ | low         | 1          |
| 3  | 1     | Sai Yai         | Sai Noi District                  | Nonthaburi             | 1              | เสี่ยงต่อ | low         | 1          |
| 4  | 1     | Khun Si         | Sai Noi District                  | Nonthaburi             | 1              | เสี่ยงต่อ | low         | 1          |
| 5  | 1     | Thawi Watthana  | Sai Noi District                  | Nonthaburi             | 1              | เสี่ยงต่อ | low         | 1          |
| 6  | 1     | Suan Phrik      | Phra Nakhon Si Ayutthaya District | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |
| 7  | 1     | Ban Khlang      | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |
| 8  | 1     | Namtao          | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |
| 9  | 1     | Wat Taku        | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |
| 10 | 1     | Bang Luang      | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |
| 11 | 1     | Bang Luang Dot  | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |
| 12 | 1     | Bang Hak        | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |
| 13 | 1     | Ban Kum         | Bang Ban District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |
| 14 | 1     | Khwan Mueang    | Bang Pahan District               | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |
| 15 | 1     | Ban Li          | Bang Pahan District               | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |
| 16 | 1     | Ban Khae        | Phak Hai District                 | Phra Nakhon Si Ayudhya | 1              | เสี่ยงต่อ | low         | 1          |

# Query to transform data type

# Batch : Screenshots or tables of sample processed data



Query results | Query stats

Completed Time in queue: 103 ms Run time: 435 ms Data scanned: 60.91 KB

Results (10)

Copy Download results CSV

Search rows

| #  | sub_district_en | province_en            | high_risk_count | medium_risk_count | low_risk_count | historical_risk |
|----|-----------------|------------------------|-----------------|-------------------|----------------|-----------------|
| 1  | Sai Noi         | Nonthaburi             | 0               | 3                 | 4              | 6               |
| 2  | Sai Yai         | Nonthaburi             | 0               | 3                 | 4              | 8               |
| 3  | Suan Phrik      | Phra Nakhon Si Ayudhya | 1               | 2                 | 4              | 10              |
| 4  | Lat Chit        | Phra Nakhon Si Ayudhya | 2               | 1                 | 6              | 12              |
| 5  | Na Khok         | Phra Nakhon Si Ayudhya | 2               | 1                 | 6              | 12              |
| 6  | Hua Wiang       | Phra Nakhon Si Ayudhya | 2               | 1                 | 6              | 12              |
| 7  | Tao Lao         | Phra Nakhon Si Ayudhya | 2               | 1                 | 5              | 12              |
| 8  | Champa Lo       | Ang Thong              | 2               | 1                 | 5              | 12              |
| 9  | Tri Narong      | Ang Thong              | 1               | 2                 | 3              | 12              |
| 10 | Pa Mok          | Ang Thong              | 2               | 1                 | 5              | 12              |

The processed data of  
risking flooding area for  
Risk Clustering Analysis.

# Batch : Screenshots or tables of sample processed data



Query results    Query stats

Completed    Time in queue: 70 ms    Run time: 527 ms    Data scanned: 137.60 KB

Results (10)

Search rows

| #  | month | province_en | district_en             | sub_district_en | avg_risk_score |
|----|-------|-------------|-------------------------|-----------------|----------------|
| 1  | 6     | Loei        | Tha Li District         | A Hi            | 1.0            |
| 2  | 2     | Yala        | Raman District          | A Song          | 1.0            |
| 3  | 11    | Yala        | Raman District          | A Song          | 1.0            |
| 4  | 12    | Yala        | Raman District          | A Song          | 3.0            |
| 5  | 7     | Loei        | Tha Li District         | A Hi            | 1.0            |
| 6  | 11    | Pattani     | Mueang Pattani District | A Noru          | 1.0            |
| 7  | 12    | Pattani     | Mueang Pattani District | A Noru          | 1.0            |
| 8  | 7     | Surin       | Buachet District        | A Phon          | 1.0            |
| 9  | 9     | Surin       | Buachet District        | A Phon          | 1.0            |
| 10 | 1     | Yala        | Raman District          | A Song          | 2.0            |

The processed data of average risking flooding score for Monthly Risk Trend Analysis.

# Streaming : Information for processing data

```
{  
  "document": "https://app.rid.go.th/reservoir/api/document/dam",  
  "date": "2025-03-27",  
  "total": 35,  
  "data": [  
    {  
      "region": "ภาคเหนือ",  
      "dam": [  
        {  
          "id": "100104",  
          "name": "เขื่อนแม่กวางอุดมธารา",  
          "owner": "กรมชลประทาน",  
          "capacity": 295,  
          "storage": 263,  
          "active_storage": 249,  
          "dead_storage": 14,  
          "volume": 151.61,  
          "percent storage": 57.65,  
          "inflow": 0.13,  
          "outflow": 0.62  
        },  
        {  
          "id": "100105",  
          "name": "เขื่อนกีวัลມ",  
          "owner": "กรมชลประทาน",  
          "capacity": 106.22,  
          "storage": 106.22,  
          "active_storage": 102.67,  
          "dead_storage": 3.55,  
          "volume": 46.89,  
          "percent storage": 44.14.  
          "inflow": 0.07,  
          "outflow": 2.24  
        }  
      ]  
    }  
  ]  
}
```

The processed data of reservoir and rainfalls for Risk Analysis of Reservoir Overcapacity.

# Streaming : Information for processing data

```
{  
  "document": "https://app.rid.go.th/reservoir/api/document/dam",  
  "date": "2025-03-27",  
  "total": 35,  
  "data": [  
    {  
      "region": "ภาคเหนือ",  
      "dam": [  
        {  
          "id": "100104",  
          "name": "เขื่อนแม่กว่างอุดมธารา",  
          "owner": "กรมชลประทาน",  
          "capacity": 295,  
          "storage": 263,  
          "active_storage": 249,  
          "dead_storage": 14,  
          "volume": 151.61,  
          "percent_storage": 57.65,  
          "inflow": 0.13,  
          "outflow": 0.62  
        },  
        {  
          "id": "100105",  
          "name": "เขื่อนก้าวม",  
          "owner": "กรมชลประทาน",  
          "capacity": 106.22,  
          "storage": 106.22,  
          "active_storage": 102.67,  
          "dead_storage": 3.55,  
          "volume": 46.89,  
          "percent_storage": 44.14,  
          "inflow": 0.07,  
          "outflow": 2.24  
        }  
      ]  
    }  
  ]  
}
```

The processed data of reservoir for Forecasting of Reservoir Water Discharge.

# Streaming : Information for processing data

```
{  
  "document": "https://app.rid.go.th/reservoir/api/document/dam",  
  "date": "2025-03-27",  
  "total": 35,  
  "data": [  
    {  
      "region": "ภาคเหนือ",  
      "dam": [  
        {  
          "id": "100104",  
          "name": "เขื่อนแม่กวางอุดมธารา",  
          "owner": "กรมชลประทาน",  
          "capacity": 295,  
          "storage": 263,  
          "active_storage": 249,  
          "dead_storage": 14,  
          "volume": 151.61,  
          "percent_storage": 57.65,  
          "inflow": 0.13,  
          "outflow": 0.62  
        },  
        {  
          "id": "100105",  
          "name": "เขื่อนกีวัฒ",  
          "owner": "กรมชลประทาน",  
          "capacity": 106.22,  
          "storage": 106.22,  
          "active_storage": 102.67,  
          "dead_storage": 3.55,  
          "volume": 46.89,  
          "percent_storage": 44.14,  
          "inflow": 0.07,  
          "outflow": 2.24  
        }  
      ]  
    }  
  ]  
}  
  
  "measurementResults": [  
    {  
      "measureTime": "2025-05-02T07:00:00",  
      "createTime": "2025-05-02T07:00:33",  
      "updateTime": "2025-05-02T07:00:33",  
      "variable": "Rainfall",  
      "value": 0.5,  
      "uom": "mm",  
      "qualityFlag": "U",  
      "comment": "No quality control",  
      "qualityControlLevel": "1"  
    }  
  ]  
},  
]  
}
```

The processed data of reservoir and rainfalls for Daily Change Analysis.